```
// Computer Program Listing Appendix Under 37 CFR 1.52(e)
// Copyright (c) 2004, Borland Software Corporation. All Rights Reserved.
procedure ReflectionReader.ReadPackage(Package: System.Type; Model:
TMoldModel);
var
  attrs: array of Attribute;
  i: integer;
  UMLAttr: UmlMetaAttributeAttribute;
  TV: UmlTaggedValueAttribute;
  nestedClasses: array of System.Type;
begin
  attrs := Attribute.GetCustomAttributes(Package,
typeof(UmlMetaAttributeAttribute));
  for i := 0 to Length(attrs)-1 do
  begin
    UMLAttr := UmlMetaAttributeAttribute(attrs[i]);
    if UMLAttr.Name = 'ownedElement' then // do not localize
      GetEnsuredClass(System.Type(UMLAttr.Value), Model)
    else
    ;
  end;
  attrs := Attribute.GetCustomAttributes(Package,
typeof(UmlTaggedValueAttribute));
  for i := 0 to Length(attrs)-1 do
  begin
    TV := UmlTaggedValueAttribute(attrs[i]);
    if TV.Tag = TAG_REGIONDEFINITIONS then
      Model.BoldTVByName[TAG_REGIONDEFINITIONS] :=
Model.BoldTVByName[TAG_REGIONDEFINITIONS] + TV.Value;
  end;
  // loop through elements that have no natural representation in code
  nestedClasses := Package.GetNestedTypes;
  for i := 0 to Length(nestedClasses)-1 do
    EnsureElement(nestedClasses[i], Model);
  Model.LoopBackIndexesValid := true;
end;
procedure ReflectionReader.ReadClassUMLAttributes(c: System.Type;
  aClass: TMoldClass);
var
  attrs: array of Attribute;
  i: integer;
  UMLAttr: UmlMetaAttributeAttribute;
begin
  attrs := Attribute.GetCustomAttributes(c,
typeof(UmlMetaAttributeAttribute));
  for i := 0 to Length(attrs)-1 do
  begin
    UMLAttr := UmlMetaAttributeAttribute(attrs[i]);
    if UMLAttr.Name = 'constraint' then // do not localize
      aClass.Constraints.Add(string(UMLAttr.Value))
```

```
    else
    ;
  end;
end;
function ReflectionReader.ConvertClass(c: System.Type; Model:
TMoldModel): TMoldClass;
var
  pi: array of PropertyInfo;
  aClass: TMoldClass;
  i: Integer;
  ElementAttr: UmlElementAttribute;
begin
  aClass := TMoldClass.Create(Model, c.Name);
  ElementAttr := UmlElementAttribute(Attribute.GetCustomAttribute(c,
typeof(UmlElementAttribute)));
  if assigned(ElementAttr) and (ElementAttr.MetaType =
'AssociationClass') then  // do not localize
    aClass.Association := GetEnsuredAssociation(c, Model);
  ConvertElement(c, aClass);
  ReadClassUMLAttributes(c, aClass);
  aClass.IsAbstract := c.IsAbstract;
  aClass.SuperClass := GetEnsuredClass(c.BaseType, Model);
  aClass.ObjectType := c;
  pi := c.GetProperties(BindingFlags(54) {BindingFlags.Instance |
BindingFlags.Public | BindingFlags.NonPublic |
BindingFlags.DeclaredOnly});
  for i := 0 to Length(pi)-1 do
  begin
    if assigned(Attribute.GetCustomAttribute(pi[i],
typeof(UmlElementAttribute))) then
      ConvertProperty(pi[i], aClass);
  end;
  result := aClass;
end;
function ReflectionReader.ConvertProperty(p: PropertyInfo; aClass:
TMoldClass): TMoldMember;
var
  RelatedClass: TMoldClass;
  aRole: TMoldRole;
  anAttr: TMoldAttribute;
  CollectionAttr: UmlCollectionAttribute;
  ElementAttr: UmlElementAttribute;
  IsRole: Boolean;
begin
  IsRole := false;
  RelatedClass := GetEnsuredClass(p.PropertyType, aClass.Model);
  if not assigned(RelatedClass) then
  begin
    CollectionAttr :=
UmlCollectionAttribute(Attribute.GetCustomAttribute(p.PropertyType,
```

```
typeof(UmlCollectionAttribute)));
    if assigned(CollectionAttr) then
      RelatedClass := GetEnsuredClass(CollectionAttr.ElementType,
aClass.Model);
    if not assigned(RelatedClass) then
    begin
      ElementAttr := UmlElementAttribute(Attribute.GetCustomAttribute(p,
typeof(UmlElementAttribute)));
      IsRole := assigned(ElementAttr) and (ElementAttr.MetaType =
'AssociationEnd');
    end;
  end;
end;
```